6.231 DYNAMIC PROGRAMMING

LECTURE 11

LECTURE OUTLINE

- Review of stochastic shortest path problems
- Computational methods for SSP
  - Value iteration
  - Policy iteration
  - Linear programming
- Computational methods for discounted problems

STOCHASTIC SHORTEST PATH PROBLEMS

- Assume finite-state system: States $1, \ldots, n$ and special cost-free termination state t
  - Transition probabilities $p_{ij}(u)$
  - Control constraints $u \in U(i)$ (finite set)
  - Cost of policy $\pi = \{\mu_0, \mu_1, \ldots\}$ is

$$J_\pi(i) = \lim_{N \to \infty} E\left\{ \sum_{k=0}^{N-1} g\big(x_k, \mu_k(x_k)\big) \,\Big|\, x_0 = i \right\}$$

  - Optimal policy if $J_\pi(i) = J^*(i)$ for all i.
  - Special notation: For stationary policies $\pi = \{\mu, \mu, \ldots\}$, we use $J_\mu(i)$ in place of $J_\pi(i)$.

- Assumption (Termination inevitable): There exists integer m such that for every policy and initial state, there is positive probability that the termination state will be reached after no more that m stages; for all π, we have

$$\rho_\pi = \max_{i=1,\ldots,n} P\{x_m \neq t \mid x_0 = i, \pi\} < 1$$

MAIN RESULT

- Given any initial conditions $J_0(1), \ldots, J_0(n)$, the sequence $J_k(i)$ generated by value iteration

$$J_{k+1}(i) = \min_{u \in U(i)} \left[g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right], \ \forall \, i$$

converges to the optimal cost $J^*(i)$ for each i.

- Bellman's equation has $J^*(i)$ as unique solution:

$$J^*(i) = \min_{u \in U(i)} \left[g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J^*(j) \right], \ \forall \, i$$

- For a stationary policy μ, $J_\mu(i)$, $i = 1, \ldots, n$, are the unique solution of the linear system of n equations

$$J_\mu(i) = g\big(i, \mu(i)\big) + \sum_{j=1}^{n} p_{ij}\big(\mu(i)\big) J_\mu(j), \quad \forall \, i = 1, \ldots, n$$

- A stationary policy μ is optimal if and only if for every state i, $\mu(i)$ attains the minimum in Bellman's equation.

BELLMAN'S EQ. FOR A SINGLE POLICY

- Consider a stationary policy μ
- $J_\mu(i)$, $i = 1, \ldots, n$, are the unique solution of the linear system of n equations

$$J_\mu(i) = g\big(i, \mu(i)\big) + \sum_{j=1}^{n} p_{ij}\big(\mu(i)\big) J_\mu(j), \quad \forall\, i = 1, \ldots, n$$

- The equation provides a way to compute $J_\mu(i)$, $i = 1, \ldots, n$, but the computation is substantial for large n $[O(n^3)]$
- For large n, value iteration may be preferable. (Typical case of a large linear system of equations, where an iterative method may be better than a direct solution method.)
- For VERY large n, exact methods cannot be applied, and approximations are needed. (We will discuss these later.)

POLICY ITERATION

- It generates a sequence $\mu^1, \mu^2, \ldots$ of stationary policies, starting with any stationary policy μ^0.

- At the typical iteration, given μ^k, we perform a policy evaluation step, that computes the $J_{\mu^k}(i)$ as the solution of the (linear) system of equations

$$J(i) = g\big(i, \mu^k(i)\big) + \sum_{j=1}^{n} p_{ij}\big(\mu^k(i)\big) J(j), \quad i = 1, \ldots, n,$$

in the n unknowns $J(1), \ldots, J(n)$. We then perform a policy improvement step,

$$\mu^{k+1}(i) = \arg \min_{u \in U(i)} \left[g(i, u) + \sum_{j=1}^{n} p_{ij}(u) J_{\mu^k}(j) \right], \ \forall \ i$$

- Terminate when $J_{\mu^k}(i) = J_{\mu^{k+1}}(i)$ $\forall$ i. Then $J_{\mu^{k+1}} = J^*$ and μ^{k+1} is optimal, since

$$\begin{aligned} J_{\mu^{k+1}}(i) &= g(i, \mu^{k+1}(i)) + \sum_{j=1}^{n} p_{ij}(\mu^{k+1}(i)) J_{\mu^{k+1}}(j) \\ &= \min_{u \in U(i)} \left[g(i, u) + \sum_{j=1}^{n} p_{ij}(u) J_{\mu^{k+1}}(j) \right] \end{aligned}$$

JUSTIFICATION OF POLICY ITERATION

- We can show that $J_{\mu^k}(i) \geq J_{\mu^{k+1}}(i)$ for all i, k
- Fix k and consider the sequence generated by

$$J_{N+1}(i) = g\big(i, \mu^{k+1}(i)\big) + \sum_{j=1}^{n} p_{ij}\big(\mu^{k+1}(i)\big) J_N(j)$$

where $J_0(i) = J_{\mu^k}(i)$. We have

$$J_0(i) = g\big(i, \mu^k(i)\big) + \sum_{j=1}^{n} p_{ij}\big(\mu^k(i)\big) J_0(j)$$

$$\geq g\big(i, \mu^{k+1}(i)\big) + \sum_{j=1}^{n} p_{ij}\big(\mu^{k+1}(i)\big) J_0(j) = J_1(i)$$

- Using the monotonicity property of DP,

$$J_0(i) \geq J_1(i) \geq \cdots \geq J_N(i) \geq J_{N+1}(i) \geq \cdots, \qquad \forall\ i$$

Since $J_N(i) \to J_{\mu^{k+1}}(i)$ as $N \to \infty$, we obtain policy improvement, i.e.

$$J_{\mu^k}(i) = J_0(i) \geq J_{\mu^{k+1}}(i) \qquad \forall\ i, k$$

- A policy cannot be repeated (there are finitely many stationary policies), so the algorithm terminates with an optimal policy

LINEAR PROGRAMMING

- We claim that J^* is the "largest" J that satisfies the constraint

$$J(i) \leq g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J(j), \qquad (1)$$

for all $i = 1, \ldots, n$ and $u \in U(i)$.

- Proof: If we use value iteration to generate a sequence of vectors $J_k = \big(J_k(1), \ldots, J_k(n)\big)$ starting with a J_0 that satisfies the constraint, i.e.,

$$J_0(i) \leq \min_{u \in U(i)} \left[g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J_0(j) \right], \quad \forall\ i$$

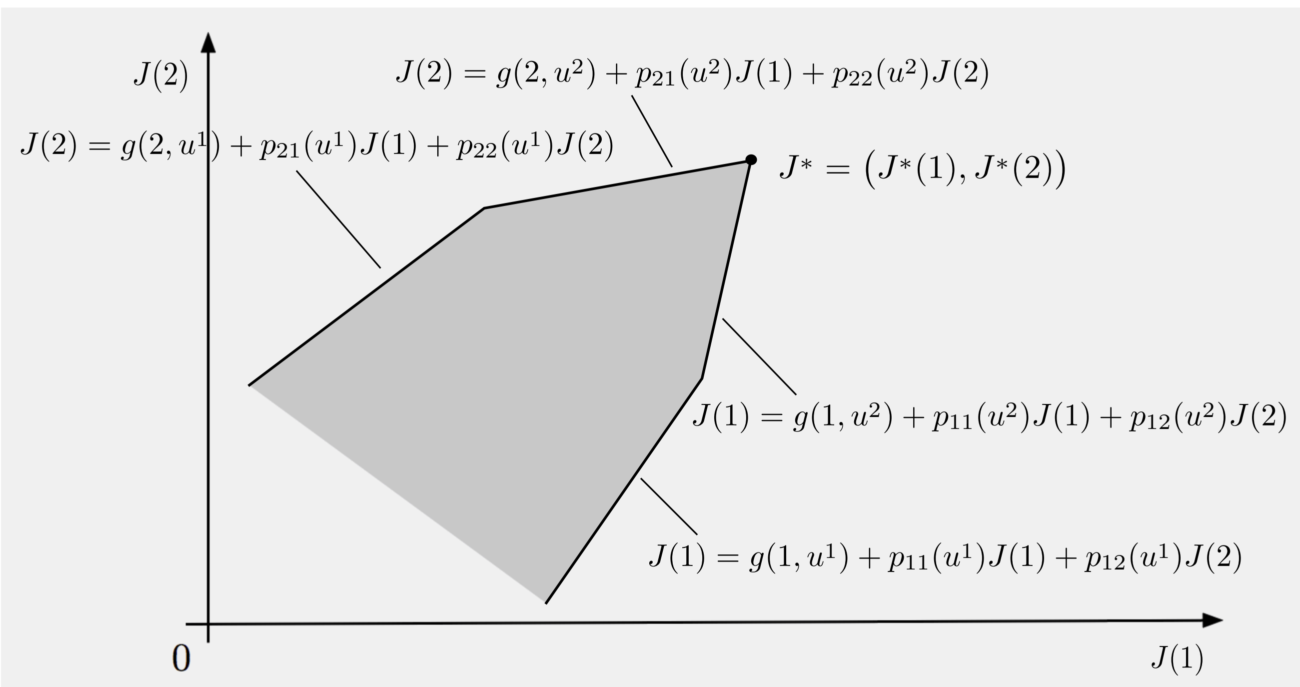
then, $J_k(i) \leq J_{k+1}(i)$ for all k and i (monotonicity property of DP) and $J_k \to J^*$, so that $J_0(i) \leq J^*(i)$ for all i.

- So $J^* = \big(J^*(1), \ldots, J^*(n)\big)$ is the solution of the linear program of maximizing $\sum_{i=1}^{n} J(i)$ subject to the constraint (1).

LINEAR PROGRAMMING (CONTINUED)

- Obtain J^* by Max $\sum_{i=1}^{n} J(i)$ subject to

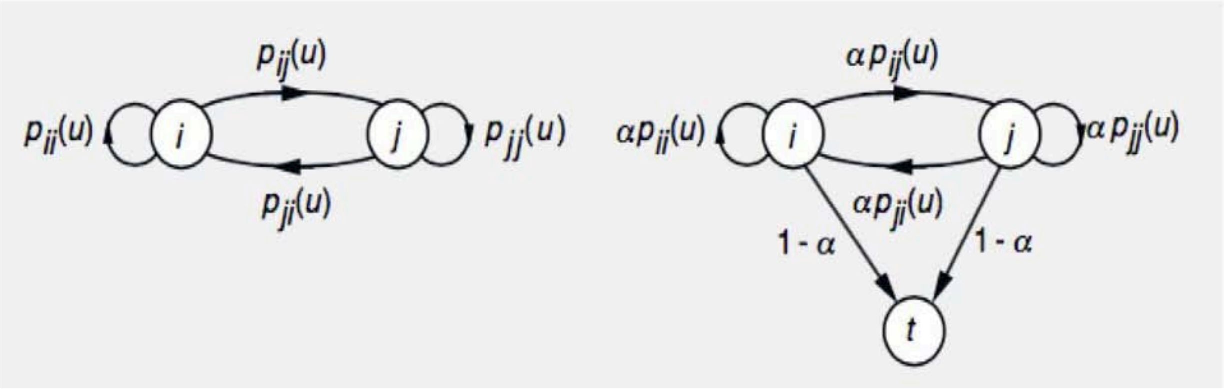
$$J(i) \leq g(i,u) + \sum_{j=1}^{n} p_{ij}(u) J(j), \quad i = 1, \ldots, n, \; u \in U(i)$$



- Drawback: For large n the dimension of this program is very large. Furthermore, the number of constraints is equal to the number of state-control pairs.

DISCOUNTED PROBLEMS

- Assume a discount factor $\alpha < 1$.
- Conversion to an SSP problem.



- kth stage cost is the same for both problems
- Value iteration converges to J^* for all initial J_0:

$$J_{k+1}(i) = \min_{u \in U(i)} \left[g(i,u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J_k(j) \right], \quad \forall \ i$$

- J^* is the unique solution of Bellman's equation:

$$J^*(i) = \min_{u \in U(i)} \left[g(i,u) + \alpha \sum_{j=1}^{n} p_{ij}(u) J^*(j) \right], \quad \forall \ i$$

- Policy iteration terminates with an optimal policy, and linear programming works.

DISCOUNTED PROBLEM EXAMPLE

- A manufacturer at each time:
  - Receives an order with prob. p and no order with prob. $1-p$.
  - May process all unfilled orders at cost $K > 0$, or process no order at all. The cost per unfilled order at each time is $c > 0$.
  - Maximum number of orders that can remain unfilled is n.
  - Find a processing policy that minimizes the α-discounted cost per stage.
  - State: Number of unfilled orders at the start of a period ($i = 0, 1, \ldots, n$).
- Bellman's Eq.:

$$J^*(i) = \min\Big[K + \alpha(1-p)J^*(0) + \alpha p J^*(1), \\ ci + \alpha(1-p)J^*(i) + \alpha p J^*(i+1)\Big],$$

for the states $i = 0, 1, \ldots, n-1$, and

$$J^*(n) = K + \alpha(1-p)J^*(0) + \alpha p J^*(1)$$

for state n.

- Analysis: Argue that $J^*(i)$ is mon. increasing in i, to show that the optimal policy is a threshold policy.